International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 4, Issue 2, Apr 2014, 43-50

TRANS STELLAR
• Journal Publications • Research Consultancy

PREVENTING BOTNETS BY SPOT FILTER METHOD

DNYANESHWAR SHINDE & PRASHANT SAWANT

Department of Computer Engineering, Pillai HOC College of Engineering & Technology, Maharashtra, India

ABSTRACT

A major security challenge on the Internet is the existence of the large number of compromised machines. Such machines have been increasingly used to launch various security attacks including DDoS, spamming, and identity theft. Two natures of the compromised machines on the Internet sheer volume and wide spread—render many existing security Countermeasures less effective and defending attacks involving compromised machines extremely hard. On the other hand, identifying and cleaning compromised machines in a network remain a significant challenge for system administrators of networks of all sizes. In this thesis we focus on the subset of compromised machines that are used for sending spam messages, which are commonly referred to as spam zombies. Given that spamming provides a critical economic incentive for the controllers of the compromised machines to recruit these machines, it has been widely observed that many compromised machines are involved in spamming.

KEYWORDS: Spam Zombies, Compromised Machines, Botnet, Spot Filter

1. INTRODUCTION

This paper studies the *network-level* behavior of spammers, including: IP address and Compromised Machines Detecting ranges that send the most spam, common spamming modes how persistent across time each spamming host is, and characteristics of spamming bot nets. We try to answer these questions by analyzing a 17-month trace of over 10 million spam messages collected at an Internet "spam sinkhole", and by correlating this data with the results of IP-based blacklist lookups, passive TCP fingerprinting information, routing information, and bot net "command and control" traces. We find that most spam is being sent from a few regions of IP address space, and that spammers appear to be using transient

"bots" that send only a few pieces of email over very short periods of time. Finally, a small, yet non-negligible, amount of spam is received from IP addresses that correspond to short-lived BGP routes, typically for hijacked prefixes. These trends suggest that developing algorithms to identify bot net membership, filtering email messages based on *network-level* properties (which are less variable than email content), and improving the security of the Internet routing infrastructure, may prove to be extremely effective for combating spam. Bot nets are becoming one of the most serious threats to Internet security.

A Spam Zombies is a network of compromised machines under the influence of malware (spam) code. The spam zombies is commandeered by a "bot master" and utilized as "resource" or "platform" for attacks such as distributed denial-of-service (DDoS) attacks, and fraudulent activities such as spam, phishing, identity theft, and information exfiltration. As a side-product of free email services, spam has become a serious problem that afflicts every Internet user in recent years. According to Message Labs, currently over 60% email traffic is spam. Although a number of anti-spam mechanisms have been proposed and deployed to foil spammers, spam messages continue swarming into Internet users'

mailboxes. A more effective spam detection and suppression mechanism close to spam sources is critical to dampen the dramatically-grown spam volume. The term "Spam Zombies" is referred to a cluster of computers infected by a malware. A single computer of this network is called a “bot” which is derived from the word “robot”. It can be defined as a software robot that operates as an agent which is capable of performing certain acts or executing commands automatically and repeatedly. In addition, it can simulate human activities. In 1993, the first bot was created as a useful feature in Internet Relay Chat (IRC) by Jeff Fisher to automate the maintenance and administration of IRC channels. Afterwards, malicious bots started to emerge to quietly bypassing the computer firewalls and taking control of the system to carry out malicious tasks. The first malicious bot running on IRC was called Pretty Park Worm which was followed by the development of several other types of similar bots. For instance, peer-to-peer bots and HTTP-based bot nets were developed between 2000 and 2005, the most popular and effective type among others. Bots can form a network of compromised computers which can be controlled by a bot master (or “herder”) using a command and control centre. Each of the compromised computers is turned into a bot also known as a “zombie”. This occurs when a user downloads or opens malicious software.

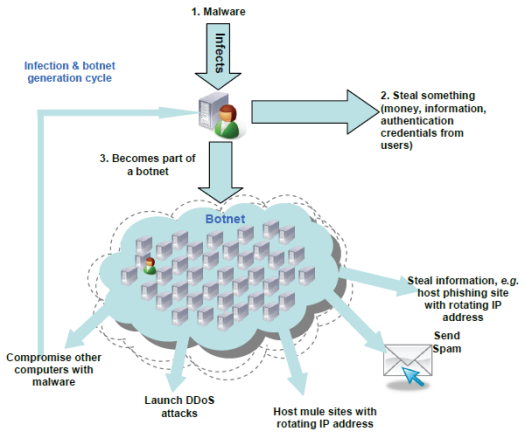

Figure 1: Botnet Lifecycle

The structure of bot net can be divided into 3 parts: (i) spammers, (ii) repeaters/relays, and (iii) backend servers. The lowest layer of this bot net structure represents the spammers which are used to recruit more bots through SPAM emails. Once a computer is infected, the bot net checks the infected host to see if it has a public IP address. If it does not have such address it becomes a spammer. That is, if it is behind a firewall, then it cannot be accessed from the Internet. On the other hand, if the infected machine has a public IP address, it is turned into a repeater. Repeaters are on the next level above the spammers. This is the entry point for any bot joining the bot net as well as the place to go for every active bot. Repeaters are considered as the middle man between the spammers and the backend servers. Requests are relayed from one layer to the next using the repeaters. Repeaters and spammers continuously exchange lists of currently active repeaters. This ensures that if a spammer lost communication with one of the repeaters (which is taken off line), it tries to communicate with the next repeater and updates the lists. In addition, repeaters exchange lists of currently active backend servers.

The list is signed with the private key of the bot net master so no other attacker can insert its backend servers and take over the bot net. Worthy of note is that other bot nets have alternative spreading techniques; e.g. Instant messaging tools such as MSN messenger, Vulnerabilities of the OS, Script injection, Peer-to-Peer sharing protocol and network shares. Figure 1 shows the bot net lifecycle from a holistic perspective. It depicts what has occurred once the infection has been made, using one of the propagation techniques discussed above. Once the victim computer gets infected with the bot,

it usually steals some data such as credentials, personal information, credit card information, etc. It then becomes part of a bot net. Once this has been accomplished, it becomes ready to perform instructed tasks such as launching a Distributed Denial of Service (DDoS) attack, compromise other computers, host a phishing website or send spam email. Bot nets can be difficult to detect, as they can run as a SMTP server mimicking a genuine server, or be a ghost proxy server tricking the defense systems as genuine services on the network. Bot nets can be silently uploaded to any system, or they can quietly spread to infect thousands of machines. This poses a serious challenge to researcher as it is not trivial to identify bot nets or locate their controller.

II. RELATED WORK

Based on email messages received at a large email service provider, two recent studies, investigated the aggregate global characteristics of spamming bot nets including the size of bot nets and the spamming patterns of bot nets. These studies provided important insights into the aggregate global characteristics of spamming bot nets by clustering spam messages received at the provider into spam campaigns using embedded URLs and near-duplicate content clustering, respectively. However, their approaches are better suited for large email service providers to understand the aggregate global characteristics of spamming bot nets instead of being deployed by individual networks to detect internal compromised machines. Moreover, their approaches cannot support the online detection requirement in the network environment considered in this paper. We aim to develop a tool to assist system administrators in automatically detecting compromised machines in their networks in an online manner.

In the following we discuss a few schemes on detecting general bot nets. Bot Hunter, developed by Gu*etal.*, detects compromised machines by correlating the IDS dialog trace in a network. It was developed based on the observation that a complete malware infection process has a number of well defined stages including inbound scanning, exploit usage, egg downloading, outbound bot coordination dialog, and outbound attack propagation. By correlating inbound intrusion alarms with outbound communications patterns, Bot Hunter can detect the potential infected machines in a network. Unlike Bot Hunter which relies on the specifics of the malware infection process, SPOT focuses on the economic incentive behind many compromised machines and their involvement in spamming.

An anomaly-based detection system named Bot Sniffer Identifies bot nets by exploring the spatial-temporal behavioral similarity commonly observed in bot nets. It focuses on IRC based and HTTP-based bot nets. In Bot Sniffer, flows are classified into groups based on the common server that they connect to. If the flows within a group exhibit behavioral similarity, the corresponding hosts involved are detected as being compromised. Bot Miner is one of the first bot net detection systems that are both protocol- and structure-independent. In Bot Miner, flows are classified into groups based on similar communication patterns and similar malicious activity patterns, respectively.

The intersection of the two groups is considered to be compromised machines. Compared to general bot net detection systems such as Bot Hunter, Bot Sniffer, and Bot Miner, SPOT is a light-weight compromised machine detection scheme, by exploring the economic incentives for attackers to recruit the large number of compromised machines. As a simple and powerful statistical method, Sequential Probability Ratio Test (SPRT) has been successfully applied in many areas. In the area of networking security, SPRT has been used to detect ports can activities, proxy-based spamming activities, anomaly-based bot net detection, and MAC protocol misbehavior in wireless networks.

III. NETWORK LEVAL MODEL

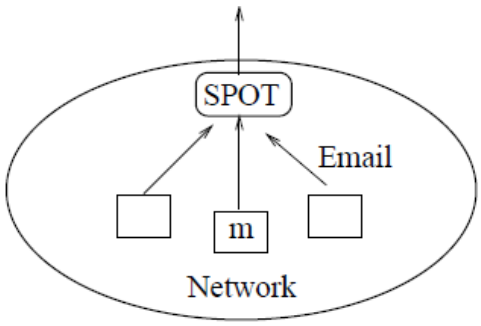

Figure 2: Network Leval Function

We assume that messages originated from machines inside the network will pass the deployed spam zombie detection system. This assumption can be achieved in a few different scenarios. First, in order to alleviate the ever-increasing spam volume on the Internet, many ISPs and networks have adopted the policy that all the outgoing messages originated from the network must be relayed by a few designated mail servers in the network. In this process the detection of incoming or outgoing message is coming from compromised or uncompromised machines will check on the network leval. Whenever any another message comes from any another network model to our network model, then with the help of SPOT filter mechanism we can find out that incoming messages are coming from compromised machines or not. SPOT filter mechanism is light weight process as compared to BOT hunter or Bot Sniffer techniques. Bot nets are now recognized as one of the most serious security threats. In contrast to previous malware, bot nets have the characteristics of command and control channel. Bot nets also often use existing common protocols, and in protocol-conforming manners. This makes the detection of bot net C&C a challenging problem. In this paper, we propose an approach that uses network-based anomaly detection to identify bot net C&C channels in a local area network without any prior knowledge of signatures or C&C server addresses.

Algorithm 1 SPOT spam zombie detection system

An outgoing message arrives at SPOT
Get IP address of sending machine m
// all following parameters specific to machine m
Let n be the message index
Let $X_n = 1$ if message is spam, $X_n = 0$ otherwise
if $(X_n == 1)$ **then**
    // spam, Eq. 3
    $\Lambda_n += ln\frac{\theta_1}{\theta_0}$
else
    // nonspam
    $\Lambda_n += ln\frac{1-\theta_1}{1-\theta_0}$
end if
if $(\Lambda_n \geq B)$ **then**
    Machine m is compromised. Test terminates for m.
else if $(\Lambda_n \leq A)$ **then**
    Machine m is normal. Test is reset for m.
    $\Lambda_n = 0$
    Test continues with new observations
else
    Test continues with an additional observation
end if

Figure 3

SPOT is designed based on the statistical tool SPRT. In the context of detecting spam zombies in SPOT, we consider $H1$ as a detection and $H0$ as normality. That is, $H1$ is true if the concerned machine is compromised, and $H0$ is

true if it is not compromised. In addition, we let $Xi = 1$ if the ith message from the concerned machine in the network is a spam, and $Xi = 0$ otherwise. Recall that SPRT requires four configurable parameters from users, namely, the desired false positive probability α, the desired false negative probability β, the probability that a message is a spam when $H1$ is true ($\mu 1$), and the probability that a message is a spam when $H0$ is true ($\mu 0$). We discuss we present the SPOT algorithm. Based on the user-specified values of α and β, the values of the two boundaries A and B of SPRT are computed using Eq. (5). In the following we describe the SPOT detection algorithm. Algorithm 1 outlines the steps of the algorithm.

When an outgoing message arrives at the SPOT detection system, the sending machine's IP address is recorded, and the message is classified as either spam or non spam by the (content-based) spam filter. For each observed IP address, SPOT maintains the logarithm value of the corresponding probability ratio An, whose value is updated according to Eq. (3) as message n arrives from the IP address (lines 6 to 12 in Algorithm 1). Based on the relation between An and A and B, the algorithm determines if the corresponding machine is compromised, normal, or a decision cannot be reached and additional observations are needed (lines 13 to 21).

We note that in the context of spam zombie detection, from the viewpoint of network monitoring, it is more important to identify the machines that have been compromised than the machines that are normal. After a machine is identified as being compromised (lines 13 and 14), it is added into the list of potentially compromised machines that system administrators

Can go after to clean. The message-sending behavior of the machine is also recorded should further analysis be required. Before the machine is cleaned and removed from the list, the SPOT detection system does not need to further monitor the message sending behavior of the machine. On the other hand, a machine that is currently normal may get compromised at a later time. Therefore, we need to continuously monitor machines that are determined to be normal by SPOT. Once such a machine is identified by SPOT, the records of the machine in SPOT are re-set, in particular, the value of ¤n is set to zero, so that a new monitoring phase starts for the machine (lines 15 to 18). SPOT requires four user-defined parameters: α, β, $\mu 1$, and $\mu 0$. In the following we discuss how a user of the SPOT algorithm configures these parameters, and how these parameters may affect the performance of SPOT. As discussed in the previous section α and β are the desired false positive and false negative rates.

They are normally small values in the range from 0.01 to 0.05, which users of SPOT can easily specify independent of the behaviors of the compromised and normal machines in the network. The values of α and β will affect the cost of the SPOT algorithm, that is, the number of observations needed for the algorithm to reach a conclusion. In general, a smaller value of α and β will require a larger number of observations for SPOT to reach detection. Ideally, $\mu 1$ and $\mu 0$ should indicate the true probability of a message being spam from a compromised machine and a normal machine, respectively. However, as we have discussed in the last section, $\mu 1$ and $\mu 0$ do not need to accurately model the behaviors of the two types of machines. Instead, as long as the true distribution is closer to one of them than another, SPRT can reach a conclusion with the desired error rates. Inaccurate values assigned to these parameters will only affect the number of observations required by the algorithm to terminate. Moreover, SPOT relies on a (content-based) spam filter to classify an outgoing message into either spam or non spam. In practice, $\mu 1$ and $\mu 0$ should model the detection rate and the false positive rate of the employed spam filter, respectively. We note that all the widely-used spam filters have a high detection rate and low false positive rate.

III. Performance of the SPOT

In this section, we evaluate the performance of SPOT based On the collected FSU emails. In all the studies, we set $=\alpha$ 0.01, β = 0.01, $\mu 1$ = 0.9, and $\mu 0$ = 0.2. That is, we assume the deployed spam filter has a 90% detection rate and 20% false positive rate. Many widely-deployed spam filters have much better performance than what we assume here.

Table 1: Perfoemance of the SPOT

Total FSU IP	Detected	Confirmed (%)	Missed (%)
440	132	126(94.7)	7(5.3)

Table 1 shows the performance of the SPOT filter. As in the table there (FSU) 440 internal IP addresses are examined in the email trace. Out of 440 Spot detected 132 were detected as associated with compromised machines. In order to understand the performance of SPOT in terms of the false positive and false negative rates, we rely on a number of ways to verify if a machine is indeed compromised. First, we check if any message sent from an IP address carries a known virus/worm attachment. If this is the case, we say we have a confirmation. Out of the 132 IP addresses identified by SPOT, we can confirm 110 of them to be compromised in this way. For the remaining 22 IP addresses, we manually examine the spam sending patterns from the IP addresses and the domain names of the corresponding machines. In the fraction of the spam messages from an IP address is high (greater than 98%), we also claim that the corresponding machine has been confirmed to be compromised.

We can confirm 16 of them to be compromised in this way. We note that the majority (62.5%) of the IP addresses confirmed by the spam percentage are dynamic IP addresses, which further indicates the likelihood of the machines to be compromised. For the remaining 6 IP addresses that we cannot confirm by either of the above means, we have also manually examined their sending patterns. We note that, they have a relatively overall low percentage of spam messages over the two month of the collection period. However, they sent substantially more spam messages towards the end of the collection period. This indicates that they may get compromised towards the end of our collection period. However, we cannot independently confirm if this is the case. Evaluating the false negative rate of SPOT is a bit tricky by noting that SPOT focuses on the machines that are potentially compromised, but not the machines that are normal. In order to have some intuitive understanding of the false negative rate of the SPOT system, we consider the machines that SPOT does not identify as being compromised at the end of the email collection period, but for which SPOT has re-set the records (lines 15 to 18 in Algorithm 1). That is, such machines have been claimed as being normal by SPOT (but have continuously been monitored). We also obtain the list of IP addresses that have sent at least a message with a virus/worm attachment. 7 of such IP addresses have been claimed as being normal, i.e., missed, by SPOT. We emphasize that the infected messages are only used to confirm if a machine is compromised in order to study the performance of SPOT. Infected messages are not used by SPOT itself. SPOT relies on the spam messages instead of infected messages to detect if a machine has been compromised to produce the results.

We make this decision by noting that, it is against the interest of a professional spammer to send spam messages with a virus/worm attachment. Such messages are more likely to be detected by anti-virus softwares, and hence deleted before reaching the intended recipients. This is confirmed by the low percentage of infected messages in the overall email trace. Infected messages are more likely to be observed during the spam zombie recruitment phase instead of spamming phase. Infected messages can be easily incorporated into the SPOT system to improve its performance. We note that both the actual false positive rate and the false negative rate are higher than the specified false positive rate and false negative

rate, respectively. One potential reason is that the underlying statistical tool SPRT assumes events (in our cases, outgoing messages) are independently and identically distributed. However, spam messages belonging to the same campaign are likely generated using the same spam template and delivered in batch; therefore, spam messages observed in time proximity may not be independent with each other. This can affect the performance of SPOT in detecting compromised machines. Another potential reason is that the evaluation was based on the FSU emails, which can only provide a partial view of the outgoing messages originated from inside FSU.

CONCLUSIONS

In this paper we present SPOT filter mechanism to prevent spam zombies. It is a very simple mechanism to detect the compromised machines that are involved in spamming activities. With the help of SPOT filter mechanism we can minimize the number of observations. This paper presents an analysis of compromised machine, their role in cyber crime and proposed SPOT filter mechanism. In this paper we performed advance analysis of various types of spam zombies and derived the appropriate use case studies.

REFERENCES

1. OECD, "Malicious Software (Malware): A SecurityThreat to the Internet Economy," 2007.
2. T. Weber, "Criminals ‘may overwhelm the web’,"*BBC News*, vol. 25, 2007.
3. P. Bacher, T. Holz, M. Kotter, and G. Wicherski. Know your enemy:Tracking botnets.
4. J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan
5. M. Xie, H. Yin, and H. Wang. An effective defense against email spam laundering. In ACM Conference on Computer and Communications *Security*, Alexandria, VA, October 30 - November 3 2006.